

LLM Cookbook 第二期

LLM的Message介绍

作者：AI匠

LLM对话系统的消息角色

- **System:** 设定上下文和规则
- **User:** 用户输入的信息
- **Assistant:** 模型生成的回复

System消息的作用

- 设定模型行为
- 提供背景信息
- 定义约束和风格
- 示例:

```
{  
  "role": "system",  
  "content": "你是一位专业的数学老师，请用简单易懂的语言回答问题。"  
}
```

DeepSeek蒸馏模型中不要使用system

所以还是要参考具体模型使用说明

1. Set the temperature within the range of 0.5-0.7 (0.6 is recommended) to prevent endless repetitions or incoherent outputs.
2. **Avoid adding a system prompt; all instructions should be contained within the user prompt.**
3. For mathematical problems, it is advisable to include a directive in your prompt such as:
"Please reason step by step, and put your final answer within `\boxed{}`."
4. When evaluating model performance, it is recommended to conduct multiple tests and average the results.

User消息的特点

- 代表用户输入的内容
- 可以是问题、指令或信息
- 示例:

```
{  
  "role": "user",  
  "content": "请解释二次方程的解法。"  
}
```

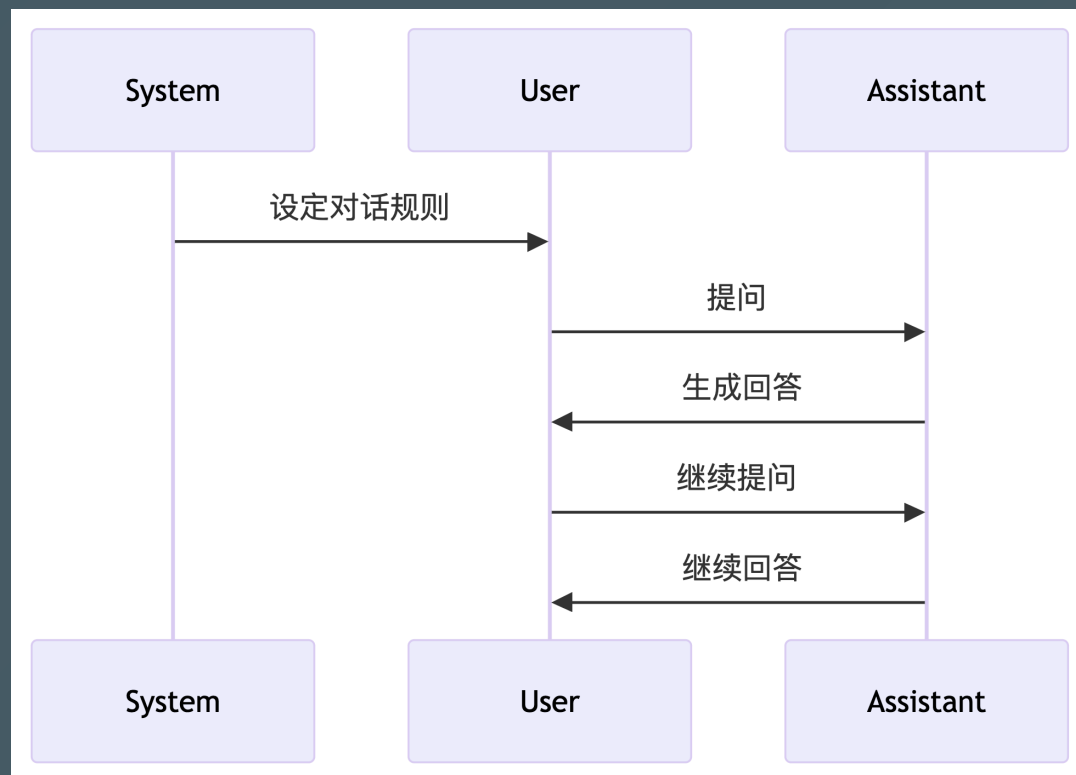
Assistant消息的特点

- 由LLM生成的回复
- 基于System和User消息
- 示例:

```
{  
  "role": "assistant",  
  "content": "二次方程的求解步骤如下..."  
}
```

消息的典型顺序

System → User → Assistant → User → Assistant ...



消息顺序的变体

- 连续**User**消息: 用于多轮用户输入
- 中途插入**System**消息: 调整模型行为
- 混合策略: 结合不同消息结构

消息位置对模型的影响

- 后面的消息影响更大
- 关键规则可放在消息末尾
- 示例:

```
[  
    {"role": "system", "content": "你是一位友善的助手。"},  
    {"role": "user", "content": "请介绍一下Python。"},  
    {"role": "system", "content": "回答时必须使用简洁风格。"}  
]
```

灵活控制System消息

- 开头放置: 设定基础规则
- 结尾放置: 强调关键指令
- 混合策略: 结合开头设定和后续调整

总结

- **LLM消息类型:** System、User、Assistant
- **System消息:** 设定规则和背景
- **User消息:** 代表用户输入
- **Assistant消息:** 生成回复
- **消息顺序影响对话:** 关键规则可放后面
- **灵活调整System消息:** 适应不同需求